

## Practice 3 Report

### Introduction:

This practice involved creating Unity scripts to control the movement and interaction of various game objects, including a cube, sphere, and cylinder. The scripts were designed to provide different functionalities and behaviors for these objects based on specific user input and physics interactions.

### Task 3: Modifying Direction of Motion

In this task, we doubled the coordinates of the direction of motion (`moveDirection`) in the `CubeMovement` script. The results showed that the cube moves in the new direction with increased speed. Doubling the coordinates effectively increased the rate of motion along the specified axes.

### Task 6: Modifying Velocity

To test the effect of changing velocity, we doubled the velocity while keeping the direction of motion the same. As expected, the cube moved faster along the same path. This demonstrates that the velocity property directly influences the speed of movement without affecting the direction.

### Task 3: Using Velocity Less Than 1

When reducing the velocity to a value less than 1 in the `CubeMovement` script, we observed that the cube moved at a slower pace. This change is consistent with the intended behavior, illustrating that velocity values below 1 result in slower movement.

### Task 3: Cube Above Y=0

We positioned the cube at a height greater than  $y=0$  and ran the `CubeMovement` script. The cube moved horizontally on the specified axes without affecting its vertical position. This demonstrates that the movement is constrained to the x and z axes, and the cube maintains its y-coordinate.

### Task 3: Local vs. World Reference Frame

We tested both local and world reference frame motions in the `CubeMovement` script. When using local coordinates, the cube moved based on its own orientation, while in the world reference frame, it moved irrespective of its orientation. This can be useful in various game scenarios, depending on the desired behavior.

### Task 12: Cylinder's Mass, Kinematics, and Friction

In this task, we controlled a cylinder's movement towards a sphere with different configurations, including changing the sphere's mass relative to the cylinder, using physics or kinematics, and adjusting friction.

- **Cylinder's Mass:** When the sphere's mass was significantly greater (10 times) than the cylinder, the cylinder struggled to move the sphere effectively. Conversely, when the cylinder's mass was greater, it could move the sphere more easily.
- **Physics vs. Kinematics:** When the cylinder was set to physics, it exhibited more realistic physical interactions, while setting it to kinematics allowed for precise, non-physical control.
- **Friction:** Doubling the friction on the cylinder led to a more rapid decrease in its velocity over time. Reducing or eliminating friction allowed the cylinder to move continuously without much slowing.

**Conclusion:**

The Unity scripts created for this project demonstrated a range of functionalities and behaviors, allowing for dynamic control and interactions within the game environment. Understanding the impact of parameters like velocity, direction, and reference frames is crucial for creating engaging and realistic gameplay experiences. Additionally, configuring objects as physical, kinematic, or trigger, and adjusting properties like mass and friction, offers flexibility in shaping in-game physics and interactions. These experiments provided valuable insights into how these factors influence gameplay mechanics.